TURCK

Industrial Automation

How to Use

Peer to Peer Communications

HOW TO

G1014
G11/12/2013

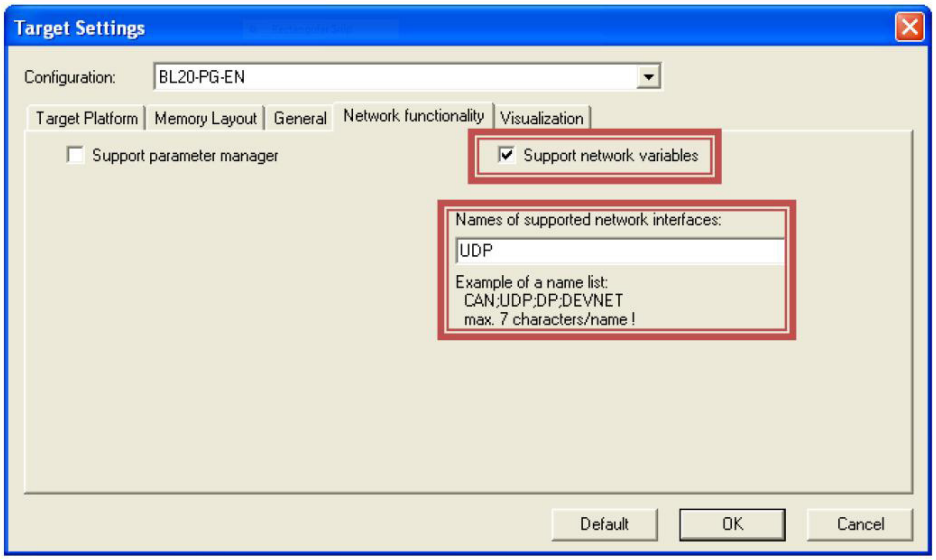
INTRODUCTION

About this Application Note

This application note explains how to establish peer-to-peer communication between TURCK programmable gateways as well as any other hardware that is programmed via CoDeSys software. The steps described in this document should be used as a guide. For further detail refer to CoDeSys help files and documentation.

Global Network Variable List

The configuration of Global Network Variable List is activated by selecting the 'Support network variables' option in the target settings.

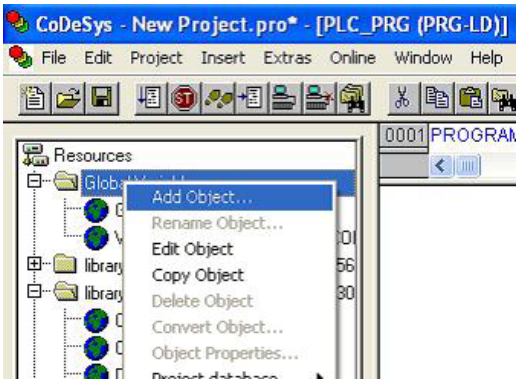


In the "Names of supported network interfaces:" type in the name of the desired networks separated by semicolons.

An example is shown below the text box. Note that the text box allows you to type in any network names, however, if the correct names are not used in this text box, you will not get the necessary settings required to setup the communication. Therefore, be cautious during this step of the process. Because this example is using the BL20-PG-EN, the UDP transmission system will be used.

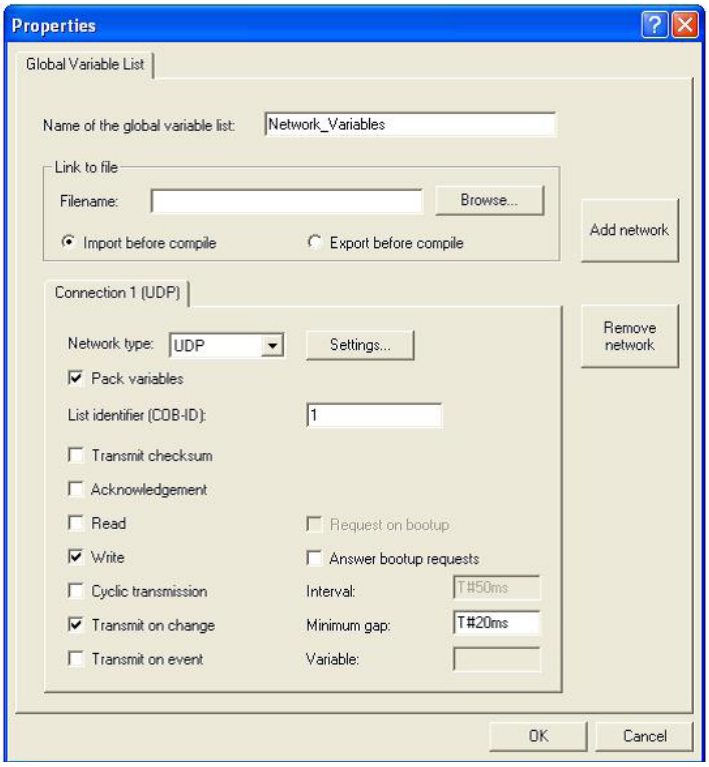
CREATING THE GLOBAL NETWORK VARIABLE LIST

1. Right-click on the "Global Variables" in the "Resources" tab and choose "Add Object…"

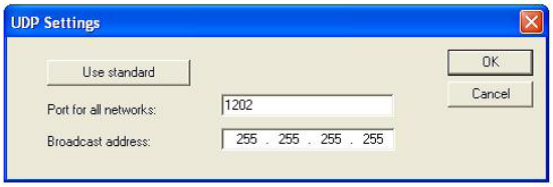


2. A "Properties" dialog box is opened. If the "Support network variables" option was selected in the target settings, an "Add network" button will be available on the right side of the dialog box. Click on the "Add network" button.

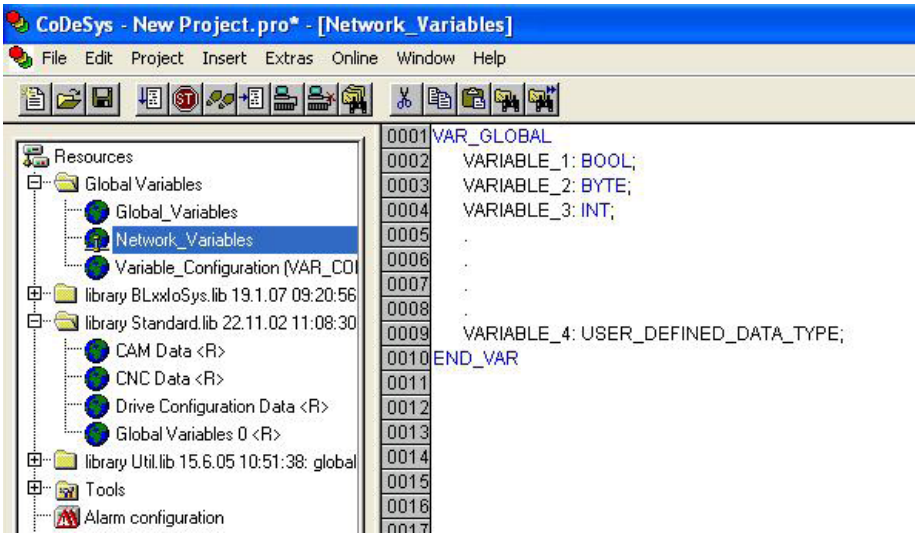
3. A "Connection 1 UDP" tab is created in the properties dialog box. Clicking on the "Add network" button again will create a "Connection 2 UDP" tab. Up to four communication tabs can be created for each list. One example of how this could be used is creating one of the connections can be configured to receive the data from a previous node by choosing the "Read" option. The other communication would send the communication to the next node by choosing the "Write" option. For detailed description of all the options inside the tab see Appendix A.



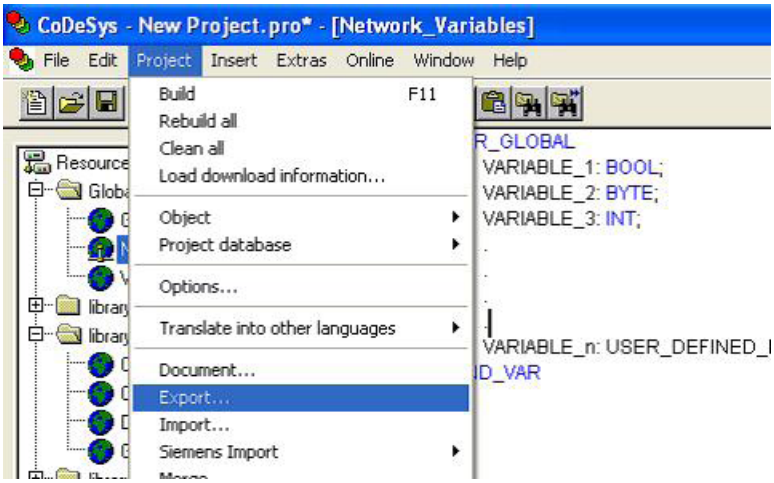
4. Click on the "Settings…" to bring up the "UDP Settings" dialog box. These settings can be modified to configure broadcast messages and node-to-node messages.



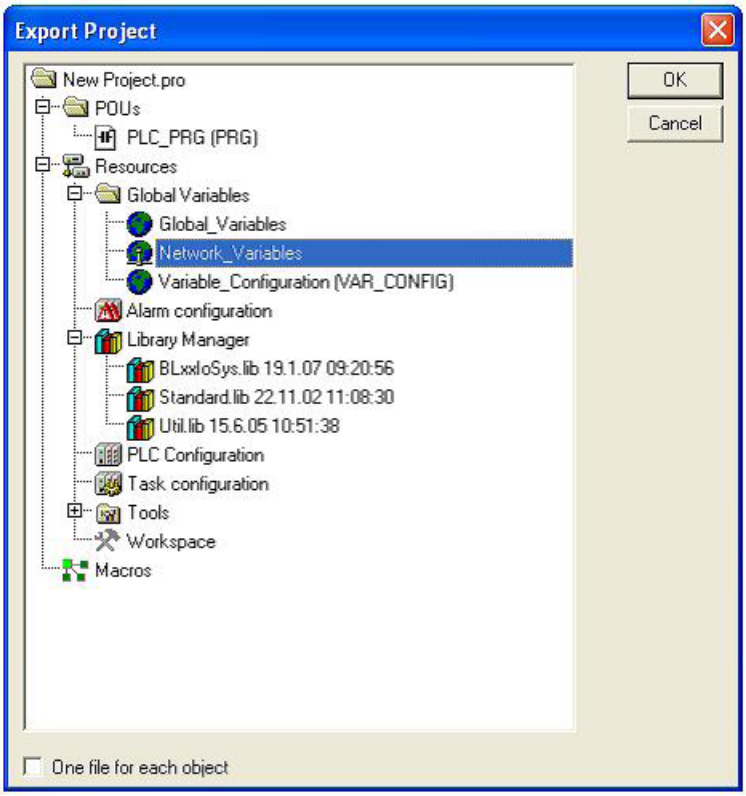
5. Type in a name in the "Name of the global variable list:" and click "OK" in the "Properties" dialog box. A list is created in the "Global Variables" folder. The Global Network Variable lists are differentiated with a icon. Open the new Global Network Variable list and create the desired variables inside the work space.



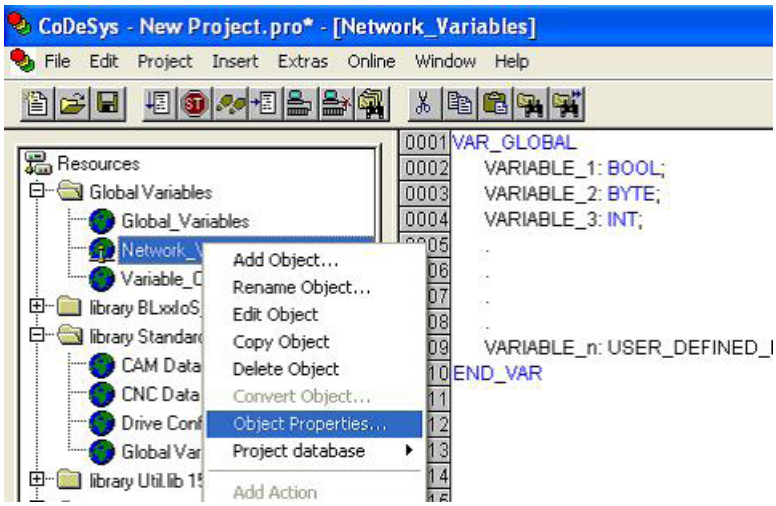
6. Configuring multiple nodes in the same network can be simplified by using the export/import functionality of CoDeSys. This will ensure that the variables in multiple nodes will match. Click on "Export…" in the "Project" menu.



7. Highlight the Global Network Variable list in the "Export Project" dialog box. If the variable list references user defined data types, it is recommended to include those data types in the export file as well.



8. The .exp file can now be imported into a new project. The communication parameters can be modified for each node by right-clicking on the variable list then choosing "Object Properties."



EXAMPLE

In this example, three gateways are configured to relay data from gateway 1 to gateway 3 through gateway 2. The data transfer from gateway 1 to gateway 2 will be done automatically on change-of-state. The data transfer from gateway 2 to gateway 3 will be triggered by another variable.

Required Hardware

- Any combination of three of the following programmable gateways.
  - BL20-PG-EN
  - BL20-PG-EN-IP
  - BL67-PG-EN
  - BL67-PG-EN-IP
- Any modules desired with corresponding bases
- Three RKM 50-*M – Power cables
  - *indicates length in meters
- Ethernet switch
- Ethernet media – at least four cables will be required. Three for the gateways and one to connect to the PC.

Required Software

The following software will be required to setup this BLident system.

- CoDeSys v 2.3.5.8 (www.turck-usa.com/Support/Downloads_~_Software/)

Create Projects

1. Assemble three programmable gateways. In this example, the IP addresses will be set to 192.168.1.x where x corresponds to the gateway number, i.e. gateway 1's IP will be 192.168.1.1, gateway 2's IP address will be 192.168.1.2, etc.
2. Create three CoDeSys projects, one for each gateway.
3. Create a Global Network Variable List in one of the projects according to the steps described above.
4. Create an .exp file of the list and import it into the other two projects.
5. Create a second communication tab in the project for gateway 2 and set the options for all projects according to the following table. Download and run the projects into the corresponding gateways.
6. Force values in gateway 1 and confirm that the values change in gateway 2
7. Trigger the transmission of data from gateway 2 to gateway 3 by forcing the "send" variable in gateway 2 or by setting the "send" variable by flagging the physical input tied to that variable and confirm that the values get written to gateway 3.

Option	Gateway 1 - Connection 1 (UDP)	Gateway 2 - Connection 1 (UDP)	Gateway 2 - Connection 2 (UDP)	Gateway 3 - Connection 1 (UDP)
Network type:*	UDP	UDP	UDP	UDP
Pack Variables	☑	☑	☑	☑
List Identifier (COB-ID):	1	1	1	1
Transmit checksum	☐	grayed out	☐	grayed out
Acknowledge	☐	grayed out	☐	grayed out
Read	☐	☑	☐	☑
Request on bootup	grayed out	☐	grayed out	☐
Write	☑	☐	☑	☐
Answer bootup requests	☐	grayed out	☐	grayed out
Cyclic transmission	☐	grayed out	☐	grayed out
Interval:	grayed out	grayed out	grayed out	grayed out
Transmit on change	☑	grayed out	☐	grayed out
Minimum gap:	T#20ms	grayed out	grayed out	grayed out
Transmit on event	unchecked	grayed out	☑	grayed out
Variable:	grayed out	grayed out	Send**	grayed out
UDP Settings				
Port for all networks:	1202	1202	1202	1202
Broadcast address:	192.168.1.2	192.168.1.1	192.168.1.3	192.168.1.2

* - "UDP" has to typed in the "Names of supported network interfaces:" field in the "Network functionality" tab in the target settings. ** - The "Send" variable needs to be declared somewhere in the project as a "BOOL." If gateway 2 has an input module, this variable can be tied to a physical input to trigger the data transmission.

WARNING AND RECOMMENDATIONS

CoDeSys allows a lot of freedom and leeway forgiveness in the setup and configuration. Several examples are:

- The Global Network Variable Lists do not necessarily need to be named the same in each node for successful communication. It seems like the number of Global Network Variable Lists in each of the nodes in the network does have to match.
- The variable names do not necessarily have to match. The order of the variables will determine how the values of the variables change.
- The data types of the variables do not need to match either. E.g. list 1 contains 4 variables of type BYTE. List 2 contains 4 variables of type BOOL. Variable x from list one can send data to variable x in list 2 as long as the value of the BYTE variable is 0 or 1. Any value greater than 1 in list 1 variables will yield an unrecognizable value in the corresponding list 2 variable.

It is recommended that great caution be exercised in setting up the Global Network Variable Lists to ensure desired performance. The following recommendation should be considered.

- Make sure that all nodes on the network have the same Global Network Variable Lists configured.
- Make sure that the parallel lists have the same amount of variable, of the same type and in the same order.
- By using the export/import functionality, these recommendation can be fulfilled.

APPENDIX A

Network type: Choose the desired type from the list. The list is defined by the target system entries. For TURCK programmable gateways the four choices available are "CAN" for CANOpen, "UDP" for a UDP transmission system, "DP" for Profibus and "DEVNET" for DeviceNet.

Settings: This button opens the dialog "Settings for <network type> with the following configuration parameters:

UDP:

Use standard: If this button is pressed, Port 1202 will be defined for the data exchange with the other network participants. The Broadcast address will be set to "255.255.255.255," which means, that the data exchange will be done with all participants in the network.

Port for all networks: Enter here a desired port number to overwrite the standard setting (see above, Use standard.) Make sure that the other nodes in the network define the same port! If you have more than one UDP connection defined in the project, then the port number will be automatically modified in all configuration sets according to the input you make here.

Broadcast address: Enter here an address of the other network node or the address range of a sub-network. (E.g. enter "192.168.1.255" if you want to communicate with all nodes with IP addresses 192.168.1.x.)

CAN:

Controller Index: Index of the CAN Controller, by which the variables should be transferred.

The following options can be activated or deactivated in configuring the transmission behavior of the variables:

Pack variables: The variables are assembled for transfer into packets (telegrams) whose size depends on the network. If the option is deactivated, a packet is setup for each variable.

List identifier (COB-ID): Identification number of the first packet in which the variables will be sent. (default = 1)

Further packets will be numbered in ascending order. Whether the network variables of the list can be defined to be "Read" and "Write" or exclusively one or the other depends on the target system. To set this property, activate the respective options "Read" and "Write."

Read: The variables in the list will be read. If the option is deactivated, further variables sent over the network will be ignored. The following options can be activated in addition:

Request on bootup: If the local node is a "Read" node (option "Read" is activated), then as soon as it gets rebooted, the actual variable values will be requested from all writing nodes and will be sent by those nodes independently of any other transmit conditions (time, event) which normally trigger the communication. Precondition: In the configuration of the writing nodes the option "Answer bootup requests" must be activated! (See below)

Write: The variables will be written. The following options can additionally be set: Transmit checksum: A checksum will be added to each packet which is sent. The checksum will be checked by the receiver to make sure that the variable definitions of sender and receiver are identical. A packet with a non-matching checksum will not be accepted, and if the "Acknowledgement" option is configured (see below), an acknowledgment of the non-matching checksum will be received.

Acknowledgment: Each message will be acknowledged by the receiver. As soon as the sender does not get at least one acknowledgement within a cycle, an error message will be produced.

Answer bootup requests: If the local node is a "Write" node (option "Write" is activated), then each bootup request sent by the reading node will be answered. That means that the actual variable values will be transmitted even if none of the other defined transmission triggers (time or event) are set.

Cyclic transmission: Variables are written within the intervals specified by the Interval: setting. (Time notation e.g. T#50ms)

Transmit on change: variables are written only when their values change. However, the value specified by the **Minimum gap:** setting can set a minimum time lapse between transfers.

Transmit on event: The variables of the list will be written when the variable specified by the Variable: setting is TRUE.